F&eIT Series

Isolated RS-422A/485 1ch
Communication Module

COM-1PD(FIT)GY

User's Manual

CONTEC CO.,LTD.

Check Your Package

Thank you for purchasing the CONTEC product.

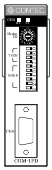
The product consists of the items listed below.

Check, with the following list, that your package is complete.   If you discover damaged or missing items, contact your retailer.

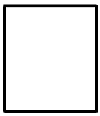
Product Configuration List

- Module[COM-1PD(FIT)GY] …1
- First step guide …1
- CD-ROM [F&eIT Series Setup Disk] *1 …1

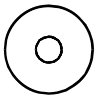
*1 The CD-ROM contains various software and User's Manual (this manual)



Module



First step guide



CD-ROM
[F&eIT Series Setup Disk]

Copyright

CONTEC CO., LTD. makes no commitment to update or keep current the information contained in this document.   The information in this document is subject to change without notice.

All relevant issues have been considered in the preparation of this document.   Should you notice an omission or any questionable item in this document, please feel free to notify CONTEC CO., LTD.

Regardless of the foregoing statement, CONTEC assumes no responsibility for any errors that may appear in this document or for results obtained by the user as a result of using this product.

Trademarks

F&eIT is a registered trademark or trademark of CONTEC CO., LTD.   Other company and product names that are referred to in this manual are generally trademarks or registered trade trademark.

Table of Contents

1. Before Using the Product

This chapter provides information you should know before using the product.

About the Module

The COM-1PD(FIT)GY performs serial communication with an external device in compliance with RS-422A/485, capable of serving as COM3 or COM4 of an F&eIT series microcontroller unit.

The module can also serve as an expansion COM port for a media converter [RP-COM(FIT)H, RP-COM(FIT)H-AF, or FX-DS540-COM2].

Please read this manual carefully to create application programs and configure the system, such as setting the switches and connecting it to external devices.

Features

- RS-422A/485 serial communication ports
- The communication lines are electrically isolated from the module.
- High-speed communication is supported at up to 921,600bps (115,200bps in compatible mode).
- A baud rate can be set by software.
- The module has 128-byte FIFO buffers for transmit and receive.
- Driver software is supplied to allow the serial ports to be used as standard Windows or Linux COM ports. *1
- The data transfer mode (full duplex or half duplex) can be set by a switch.
- The board includes a 100Ω terminating resistor required for multi-drop (party line) connections. The resistor can be inserted into the signal line by a switch.
- Surge protection is provided for each RS-422A/485 signal line.
- Up to three units can be added (at the time of enhanced mode) as expansion ports for micro controller unit. *2
- Up to three units can be added as expansion ports for media converters [RP-COM(FIT)H, RP-COM(FIT)H-AF, FX-DS540-COM2].
- Similar to other F&eIT series products, the system, in the module itself, incorporates a 35-mm DIN rail mounting mechanism as a standard item. A connection to a controller module can be effected on a lateral, stack basis in a unique configuration, which permits a simple, smart system configuration without the need for a backplane board.

*1: When using it as the expansion port for micro controller unit

*2: Two units can be added at the time of compatible mode

Customer Support

CONTEC provides the following support services for you to use CONTEC products more efficiently and comfortably.

Web Site

Japanese	http://www.contec.co.jp/
English	http://www.contec.com/
Chinese	http://www.contec.com.cn/

Latest product information

CONTEC provides up-to-date information on products.

CONTEC also provides product manuals and various technical documents in the PDF.

Free download

You can download updated driver software and differential files as well as sample programs available in several languages.

Note! For product information
Contact your retailer if you have any technical question about a CONTEC product or need its price, delivery time, or estimate information.

Limited One-Year Warranty

CONTEC products are warranted by CONTEC CO., LTD. to be free from defects in material and workmanship for up to one year from the date of purchase by the original purchaser.

Repair will be free of charge only when this device is returned freight prepaid with a copy of the original invoice and a Return Merchandise Authorization to the distributor or the CONTEC group office, from which it was purchased.

This warranty is not applicable for scratches or normal wear, but only for the electronic circuitry and original modules. The warranty is not applicable if the device has been tampered with or damaged through abuse, mistreatment, neglect, or unreasonable use, or if the original invoice is not included, in which case repairs will be considered beyond the warranty policy.

How to Obtain Service

For replacement or repair, return the device freight prepaid, with a copy of the original invoice. Please obtain a Return Merchandise Authorization Number (RMA) from the CONTEC group office where you purchased before returning any product.

* No product will be accepted by CONTEC group without the RMA number.

Liability

The obligation of the warrantor is solely to repair or replace the product. In no event will the warrantor be liable for any incidental or consequential damages due to such defect or consequences that arise from inexperienced usage, misuse, or malfunction of this device.

Safety Precautions

Understand the following definitions and precautions to use the product safely.

Safety Information

This document provides safety information using the following symbols to prevent accidents resulting in injury or death and the destruction of equipment and resources. Understand the meanings of these labels to operate the equipment safely.

⚠ DANGER	DANGER indicates an imminently hazardous situation which, if not avoided, will result in death or serious injury.
⚠ WARNING	WARNING indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury.
⚠ CAUTION	CAUTION indicates a potentially hazardous situation which, if not avoided, may result in minor or moderate injury or in property damage.

Handling Precautions

⚠ CAUTION

Take the following precautions when handling this module.

- Do not modify the module. CONTEC will bear no responsibility for any problems, etc., resulting from modifying this module.
- Do not use or store the equipment in a hot or cold place, or in a place that is subject to severe temperature changes. (Operating temperature range: 0 to 50°C)
- Do not use or store the equipment in a place subject to direct sunlight or near a heating device, such as a stove.
- Do not use or store the equipment in a dusty or humid place. (Operating humidity range: 10 to 90%RH, no condensation)
- As this product contains precision electronic components, do not use or store in environments subject to shock or vibration.
- Do not use or store the product near equipment generating a strong magnetic field or radio waves.
- If you notice any strange odor or overheating, please unplug the power cord immediately.
- In the event of an abnormal condition or malfunction, please consult the dealer from whom the equipment was purchased.
- To avoid electric shock, please do not touch the system with a wet hand.
- Do not open the module casing. CONTEC will disclaim any responsibility for equipment whose casing has been opened.
- To prevent damage, please do not subject the module to impact or bend it.
- To prevent contact malfunction, please do not touch the metallic pins on the external module connector.
- The module contains switches that need to be properly set. Before using the module, please check its switch settings.
- To avoid malfunction, please do not change the module switch settings in an unauthorized manner.
- "Do not operate the device module when the power for the Controller Module is on.
  To avoid malfunction, please be sure to turn off the power for the Controller Module."
- When disposing of the product, follow the disposal procedures stipulated under the relevant laws and municipal ordinances.

- Regarding “EMC Instruction Class A Notice and FCC Part 15 Class A Notice and VCCI Class A”
  This product has acquired the above-mentioned standard.
  However, a sufficient margin may not be secured for the standard. In this case, use a ferrite core (SEIWA E04SR301334 or a compatible product) for both ends of the COM cable.
  When attaching the ferrite core, coil it around once near the connector while leaving it open, and then close it.
  If a mouse is connected to your microcontroller, as a rider to the EMC Directive for a Class A product, a ferrite core (SEIWA E04SR301334 or a compatible product) must be used for the mouse cable.
  When attaching the ferrite core, coil it around twice near the connector while leaving it open, and then close it.

FCC PART 15 Class A Notice

NOTE

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in commercial environment.
This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference at his own expense.

WARNING TO USER

Change or modifications not expressly approved the manufacturer can void the user's authority to operate this equipment.

Environment

Use this product in the following environment. If used in an unauthorized environment, the module may overheat, malfunction, or cause a failure.

Operating temperature

0 - 50°C

Humidity

10 - 90%RH (No condensation)

Corrosive gases

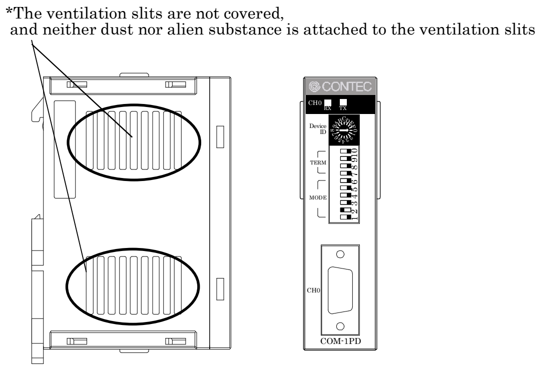
None

Floating dust particles

Not to be excessive

Inspection

Inspect the product periodically as follows to use it safely.



Storage

When storing this product, keep it in its original packing form.

(1) Put the module in the storage bag.

(2) Wrap it in the packing material, then put it in the box.

(3) Store the package at room temperature at a place free from direct sunlight, moisture, shock, vibration, magnetism, and static electricity.

Disposal

When disposing of the product, follow the disposal procedures stipulated under the relevant laws and municipal ordinances.

2. Module Nomenclature and Settings

Nomenclature of Module Components

Figure 2.1. shows the names of module components. In the figure, the indicated switch settings represent factory settings.

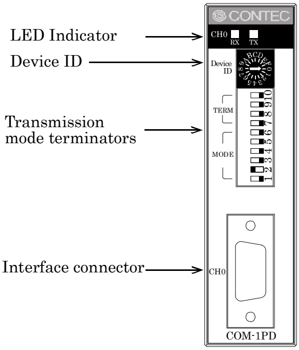


Figure 2.1. Names of module components

Setting a Device ID

Depending on the specific Device ID that is set, the COM-1PD(FIT)GY can be switched over between the compatible mode and the enhanced mode.

Table 2.1 shows the relationship between Device IDs and the modes.
For a description of the compatible and enhanced modes, see Chapter 4, "Using the Module".

The factory setting is [0], in which the COM-1PD(FIT)GY can be used as a COM3 standard port.

⚠ CAUTION

The setup method in this document assumes the combination with a microcontroller. For the combination with a media converter, read the manual for the media converter carefully for setup.

Setup Method

A device ID can be set by turning the rotary switch on the device face.
To set a device ID, turn the switch knob.

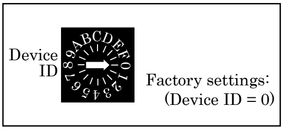


Figure 2.2. Setting a Device ID

Table 2.1. I/O Address

DeviceID	CH0		
	I/O address(h)	Interrupt	Mode
0	03E8 - 03EF	IRQ5	Compatible (COM3)
1	03E8 - 03EF	Not Used	Compatible (COM3)
2	02E8 - 02EF	IRQ7	Compatible (COM4)
3	02E8 - 02EF	Not Used	Compatible (COM4)
4	01A0 - 01A7	IRQ5	Enhanced mode
5	01A0 - 01A7	IRQ7	Enhanced mode
6	02A0 - 02A7	IRQ5	Enhanced mode
7	02A0 - 02A7	IRQ7	Enhanced mode
8	9800 - 9807	IRQ9	Enhanced mode
9	9820 - 9827	IRQ5	Enhanced mode
A	9840 - 9847	IRQ7	Enhanced mode
B	9860 - 9867	Not Used	Enhanced mode
C	0700 - 0707	IRQ9	Enhanced mode
D	0720 - 0727	IRQ7	Enhanced mode
E	Reserved	Reserved	Reserved
F	Reserved	Reserved	Reserved

⚠ CAUTION

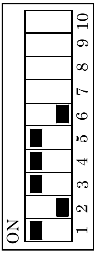
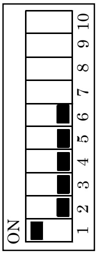
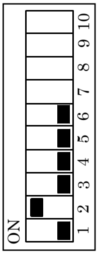
If the model label on the main unit reads "Rev.D", Device ID No. C or D can be used for that product.
Please use it excluding Device ID No. 8, 9, A, B when using in the Windows environment.

Setting Transmission Mode

The data transfer mode setting switch is used to switch between full duplex and half duplex and to specify whether to use RTS/CTS in full duplex mode. Set the appropriate data transfer mode for the device with which you are communicating. Bits 1 to 5 are used to set the data transfer mode. Always set bit 6 is to OFF.

Setting Procedure

Table 2.2.   Setting Transmission Mode

Transmission mode	Half duplex [Half]	Full duplex [Full]	
		When there are no RTS and CTS RTS ↷ CTS ←	When there are RTS and CTS RTS → CTS ←
Setting	 TXD is only used for data transmission; the sending and receiving modes should be switched over using the modem control register.	 With above setting, RTS and CTS are connected in the board. Communication is available without wiring of RTS and CTS.	 Factory setting To connect RTS and CTS to the other unit, communication is available.

⚠ CAUTION

Do not use with bits 1 and 2 both set ON as this may result in damage to the module.

Setting of Terminator

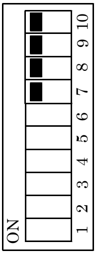
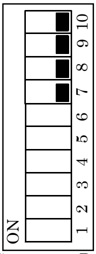
The terminator setting switch controls whether or not a terminator is inserted into each signal line. Set the terminators on or off in accordance with the devices with which you are communicating. The terminators on the module are 100Ω resistors. See Figure 2.4 for details on how to use terminators in a multi-drop (party line) connection.

Each bit in the switch corresponds to a different signal line: bit 7 for RxD, bit 8 for CTS, bit 9 for TxD, and bit 10 for RTS.

Setting Procedure

If you wish to use a terminator of other than 100Ω, set the terminator switch OFF and insert an external terminator.

Table 2.3. Setting of Terminator

	Inserted [Term ON]	Not inserted [Term OFF]
Setting	 Figure 2.3. A or D (Both terminators of the line)	 Factory setting Figure 2.3. B or C (Other than both terminators of the line)

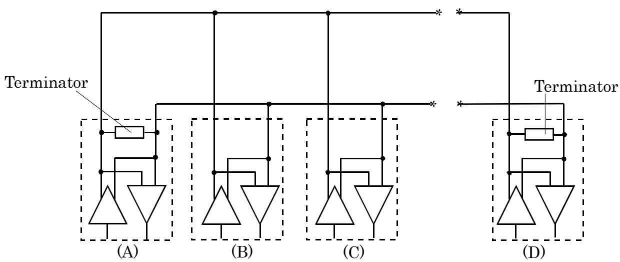


Figure 2.3. Party Line Connection

⚠ CAUTION

When the data transfer mode is set to half duplex, only set bit 9 ON. Communications may not function if other bits are set ON.

The figure below shows the circuit associated with the data transfer mode setting switch and terminator setting switch.

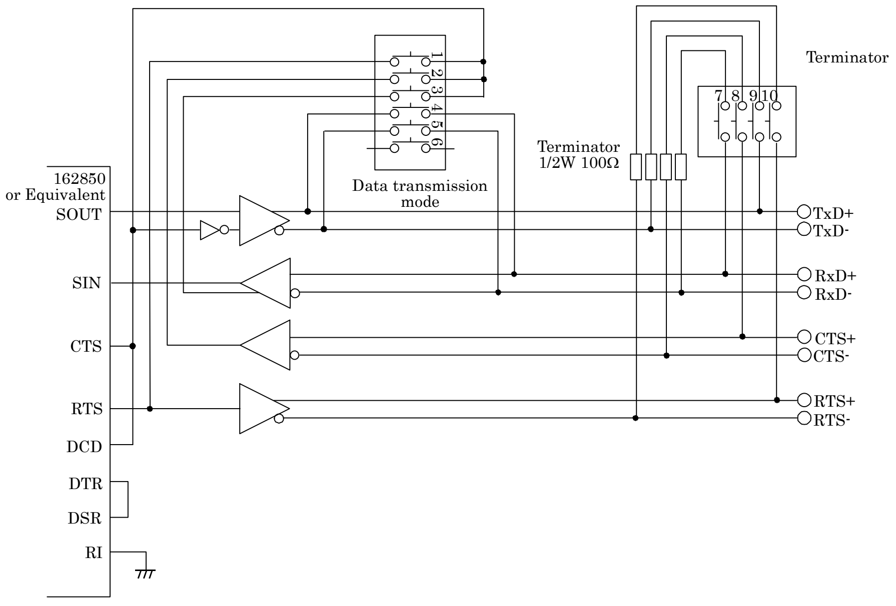


Figure 2.4. Setting Switch Circuits

LED Indicator

RX: Indicates that the module is receiving serial data (green).

TX: Indicates that the module is transmitting serial data (green).

3. Connecting to an External Device

Interface Connector

How to Connect an Interface Connector

When connecting the Module to an external device, you can use the supplied connector plug

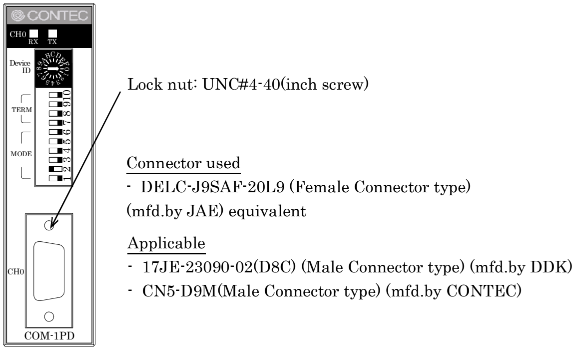


Figure 3.1. Connecting an interface connector and connectors that can be used

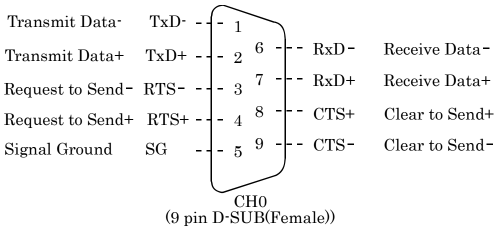


Figure 3.2. Signal Layout on the Interface Connector

⚠ CAUTION

- For TxD, RxD, and RTS, big numbered pins are + and small numbered pins are -.
  For CTS, big numbered pins are - and small numbered pins are +. This is the opposite of the other signals, but is not a misprint.
- The external metal frame (shell) of the connector is insulated from the unit, or not in contact with any part of the unit. When grounding is required, for example, as data transfer remains unstable, ground the remote device to the earth.

Types of Cable and Example Connections

The figures below show examples of how to connect the cable for the module.
The RS-422A/485 interface works based on a differential signal whereby the signal is carried by the potential difference between two lines (+ and -). Using twisted pair cable is recommended to improve resistance to noise.

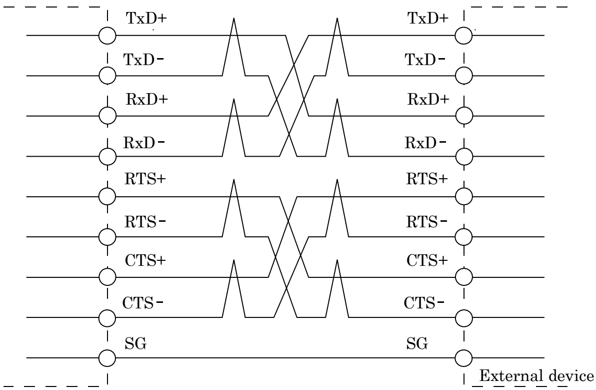


Figure 3.3. Example Connection RTS and CTS to a External Device in Full Duplex

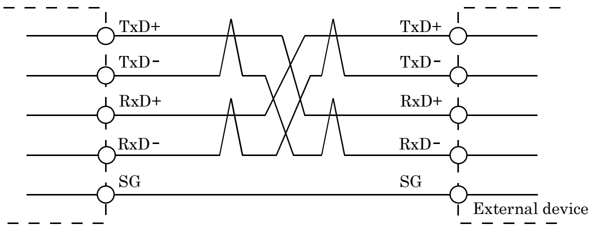


Figure 3.4. Example Connection Oneself loop to RTS and CTS in Full Duplex

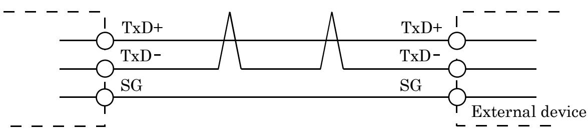


Figure 3.5. Example Connection in Half Duplex

⚠ CAUTION

If connecting between external devices and this board with faulty wiring, it will become the cause of failure.

4. Functions

This section describes the functions of the module.

Communication Function

Serial Data Transmission

Sends and receives data in accordance with the RS-422A/485 standard.

The baud rate can be set independently in the range 2 - 921,600bps by software.

RS-422A/485 Control Lines

The module include the RTS+, RTS-, CTS+, and CTS- control lines.

The lines can be controlled or monitored by software from the application

Send and Receive Data Buffers

The module has a separate 128-byte send and 128-byte receive buffer.

The buffers operate as FIFO buffers and help reduce the load on the CPU for high-speed communications or system operation.

The FIFO trigger size is a data buffer size that determines the timing of generating a hardware interrupt.

Increasing the FIFO trigger size decreases the number of times hardware interrupts occur for data transmission and reception, thereby reducing the load on the CPU.

However, this delays the timing of transmitting data to the remote device and the timing of notifying an application of data reception because actual transmission and reception take place after a certain amount of data has been collected.

In contrast, decreasing the FIFO trigger size or disabling FIFO memory shortens the timing of data transmission and reception but may increase the load on the CPU or miss some receiving data.

Since the FIFO trigger size is variable, adjusting it depending on the system results in optimum performance.

Setting the Baud Rate

The output baud rate can be set by setting the appropriate value to the Baud Rate Generator register. The available setting range depends on which clock mode is used.

Clock 3 mode: 15 - 921,600bps
Clock 2 mode: 8 - 460,800bps
Clock 1 mode: 4 - 230,400bps
Clock 0 mode: 2 - 115,200bps

However, some baud rate values do not have a corresponding Baud Rate Generator register setting. If the result of substituting the baud rate into the equation below is an integer, then that baud rate can be set. If the result contains a fractional part, the baud rate cannot be set.

Clock 3 mode

921600 ÷ Desired baud rate = Division register setting value

Ex.) 921600 ÷ 9600bps = 96 (As the result is an integer, this baud rate can be set.)

921600 ÷ 128000bps = 7.2 (As the result contains a fractional part, this baud rate cannot be set.)

Clock 2 mode

460800 ÷ Desired baud rate = Division register setting value

Ex.) 460800 ÷ 9600bps = 48 (As the result is an integer, this baud rate can be set.)

460800 ÷ 128000bps = 3.6 (As the result contains a fractional part, this baud rate cannot be set.)

Clock 1 mode

230400 ÷ Desired baud rate = Division register setting value

Ex.) 230400 ÷ 9600bps = 24 (As the result is an integer, this baud rate can be set.)

230400 ÷ 128000bps = 1.8 (As the result contains a fractional part, this baud rate cannot be set.)

Clock 0 mode

115200 ÷ Desired baud rate = Division register setting value

Ex.) 115200 ÷ 9600bps = 12 (As the result is an integer, this baud rate can be set.)

115200 ÷ 76800bps = 1.5 (As the result contains a fractional part, this baud rate cannot be set.)

⚠ CAUTION

See Chapter 5 “Interrupt Vector Registers” for selecting the clock frequency (1.8432, 3.6864, 7.3728, or 14.7456 MHz).
For use in compatible mode, clock 0 (1.8432 MHz) is used as a fixed setting.

Refer to the following baud rate setting examples. Baud rates other than those listed below can also be set if they produce a valid setting value in the equation described above.

Table 4.1. Baud Rate Generator Programming Table

Output baud rate	Clock 0 mode (1.8432MHz)		Clock 1 mode (3.6864MHz)		Clock 2 mode (7.3728MHz)		Clock 3 mode (14.7456MHz)	
	Baud Rate Generator register	Setup error (%)	Baud Rate Generator register	Setup error (%)	Baud Rate Generator register	Setup error (%)	Baud Rate Generator register	Setup error (%)
2	57600	-						
4	28800	-	57600	-				
5	23040	-	46080	-				
8	14400	-	28800	-	57600	-		
15	7680	-	15360	-	30720	-	61440	-
50	2304	-	4608	-	9216	-	18432	-
75	1536	-	3072	-	6144	-	12288	-
110	1047	0.026	2094	0.026	4189	0.0022	8378	0.0022
134.5	857	0.058	1713	0.0006	3426	0.0006	6852	0.0006
150	768	-	1536	-	3072	-	6144	-
300	384	-	768	-	1536	-	3072	-
600	192	-	384	-	768	-	1536	-
1200	96	-	192	-	384	-	768	-
1800	64	-	128	-	256	-	512	-
2000	58	0.68	115	0.17	230	0.17	461	0.04
2400	48	-	96	-	192	-	384	-
3600	32	-	64	-	128	-	256	-
4800	24	-	48	-	96	-	192	-
7200	16	-	32	-	64	-	128	-
9600	12	-	24	-	48	-	96	-
14400	8	-	16	-	32	-	64	-
19200	6	-	12	-	24	-	48	-
28800	4	-	8	-	16	-	32	-
38400	3	-	6	-	12	-	24	-
57600	2	-	4	-	8	-	16	-
76800			3	-	6	-	12	-
115200	1	-	2	-	4	-	8	-
153600					3	-	6	-
230400			1	-	2	-	4	-
460800					1	-	2	-
921600							1	-

Automatic RTS Control Functions

This function applies to half duplex mode communications.

As half duplex means that the same line is used for sending and receiving, the RTS and CTS signals are used to switch between sending and receiving. Normally, RTS is set by writing to the corresponding register bit, but on this module it is controlled by hardware. This reduces the load on the CPU.

Other Functions

Bus Isolation

The communication lines are electrically isolated from each other and from the PC.

This isolation prevents electrical disturbances from occurring between the Module and the external circuitry.

The Module can therefore be used comfortably even when line noise can be easily generated to seemingly cause the Module to malfunction or break.

Surge Protection

As surge protection is provided on all RS-422A/485 control lines, you can safely use the modules in environments where you are concerned about surges causing incorrect operation or damage to the PC.

5. Using the Module

Compatible and Enhanced Modes

The COM-1PD(FIT)GY can operate in two modes: the compatible mode, in which the COM-1PD(FIT)GY, when connected to CONTEC’s microcontroller unit, acts as a standard serial port; and the enhanced mode, in which the COM-1PD(FIT)GY operates under CONTEC’s unique control method. Before building a system using the COM-1PD(FIT)GY, a working understanding of the features of these modes may be in order.

* To use three or more channels of COM in the Windows environment, select the enhanced mode, and then use the COM-DRV(W32) driver. (Ex. : Device ID-No. 04h, 07h, 0Ch)

Compatible Mode

The COM-1PD(FIT)GY can use two channels as standard serial ports.
COM-1PD(FIT)GY can be assigned to COM3 or COM4.

The module uses the I/O addresses assigned for a standard serial port. As the module is handled as the standard serial port, it can run under other driver software that can operate standard serial ports.

Although OS-dependent, the I/O addresses in the COM-1PD(FIT)GY can be recognized by the system and used as standard ports. You can use these addressed by checking the system settings.

Enhanced Mode

The enhanced mode operates the COM-1PD(FIT)GY by using CONTEC's unique control method.

For I/O addresses, CONTEC-designated addresses must be used.

When the COM-DRV(W32) driver is used, it can be used in the same way as a standard serial port.

⚠ CAUTION

When using it in the OS other than Windows, user needs to create the device driver for enhanced mode.

Table 5.1. I/O Address

Device ID	CH0		
	Mode	I/O address(h)	Interrupt Level
0	Compatible (COM3)	03E8 - 03EF	IRQ5
1	Compatible (COM3)	03E8 - 03EF	Not Used
2	Compatible (COM4)	02E8 - 02EF	IRQ7
3	Compatible (COM4)	02E8 - 02EF	Not Used
4	Enhanced mode	01A0 - 01A7	IRQ5
5	Enhanced mode	01A0 - 01A7	IRQ7
6	Enhanced mode	02A0 - 02A7	IRQ5
7	Enhanced mode	02A0 - 02A7	IRQ7
8	Enhanced mode	9800 - 9807	IRQ9
9	Enhanced mode	9820 - 9827	IRQ5
A	Enhanced mode	9840 - 9847	IRQ7
B	Enhanced mode	9860 - 9867	Not Used
C	Enhanced mode	0700 - 0707	IRQ9
D	Enhanced mode	0720 - 0727	IRQ7
E	Reserved	Reserved	Reserved
F	Reserved	Reserved	Reserved

⚠ CAUTION

- If the model label on the main unit reads "Rev.D", Device ID No. C or D can be used for that product.
- Please use it excluding Device ID No. 8, 9, A, B when using in the Windows environment.

Operating under CPU-SB30 Series

When using the module under CONTEC’s microcontroller unit CPU-SB30 Series, the OS must be set to recognize the I/O address and interrupt level used by the COM-1PD(FIT)GY.
In CPU-SB30 Series, this is called hardware installation. Use the following installation procedure.

Compatible Mode Installation Procedure

(1) Do not connect COM-1PD(FIT)GY to CPU-SB30 Series. Turn on the CPU-SB30 Series.
(2) Select [Start] from [Control Panel] and start the [Add Hardware].
(3) Click [Next >] in response to [Welcome to the Add Hardware Wizard] in the Add Hardware Wizard.
(4) Select the [Yes] from the [Has the hardware already been connected with the computer?] screen, and click [Next].
(5) The next, select [Add a New Device] in response to [Choose a Hardware Task] and click [Next].
(6) Select [No, select from a list [Advanced]] in response to [Do you want Windows to search for your new hardware ?], and click [Next].
(7) Select the [Ports (COM & LPT)] folder from the [Hardware types:] screen.
(8) Select [Communications Port] from [Standard Types], and press [Next].
(9) On the [Start Hardware Installation] screen, press [Next].
(10) [Code 34] occurs in the [Completing the Add Hardware Wizard] screen, as resources are not assigned properly. To assign resources appropriately, select [View or change resources for this hardware (Advanced)].

(11) On the [Properties] screen of [Add New Hardware Wizard], press the [Set Configuration Manually] to specify an I/O address and interrupt level.
(12) Change [Base Configuration] and select an available I/O address. Then press [IRQ] to specify IRQ. Make sure that [Conflict information] is set to [No devices are conflicting].
(13) Make sure that [Conflicting device list] is set to [No conflicts].
   * Set the I/O address and interrupt level in the same way as for DeviceID. For details, see Table 5.1.
(14) Press [Complete] to end the wizard.
(15) The message [Do you want to restart now?] will appear. Select [Yes] to restart the OS.
(16) Once the operation is completed according to the instructions, the installation will be completed. When using more than one unit in Compatible Mode, repeat (2) - (15).
(17) Set DeviceID according to the I/O address and interrupt level registered in the system.
(18) Always make sure that CPU-SB30 Series is turned off, when connecting COM-1PD(FIT)GY to CPU-SB30 Series.
(19) Turn on CPU-SB30 Series to start the OS. This completes the installation. After the completion of the installation, always check the resources, using the [How to Verify Resources Managed by OS] section as a reference.

Enhanced Mode Installation Procedure

When it is used with Enhanced Mode, COM driver of the bundled CD-ROM is necessary.
Refer to "COMDRV(W32)Main_e.html" being attached to the bundled CD-ROM for the way of installing it.

⚠ CAUTION

When using it with Enhanced Mode in the Windows, the product cannot be used with Device ID 4 or 5. To use the product with Enhanced Mode, select another Device ID.

How to Verify Resources Managed by OS

Always check the PC resources (I/O address and interrupt level) assigned to the COM-1PD(FIT)GY before actually using this product. Use the following procedure to check the resources managed by the OS.

(1) Select [System] from [Control Panel] and click [Hardware] property sheet, then open [Device Manager].
(2) For Compatible mode, double click on the [Ports (COM & LPT)] folder. For Enhanced mode, double click on the [Multi-function adapters] folder.
(3) Double click on the [CONTEC CO., LTD. - COM-1PD(FIT)GY] or [COM*] folder to display the properties screen.
(4) Select [Resources]. Check the resource items and settings, and look for any conflicts.
(5) If changing an I/O address, change the Basic configuration from the [Setting based on:]. To change an interrupt level, click on [Change setting (C)].

After checking the resources, check again that the interrupt level value on the COM-1PD(FIT)GY match the settings in the OS.

Checking Operations with the Diagnosis Program method

Use the Serial diagnosis program from the supplied CD-ROM API-SBP(W32) to check that the module and driver software work normally, thereby you can confirm that they have been set up correctly.

What is the Diagnosis Program?

These programs perform some simple checks on the board operation. Two programs are provided.

Terminal program (CTstCom.exe)

Data entered from the keyboard is sent directly from the port. The function of the program is equivalent to the Hyper Terminal program provided with Windows.

Serial Communications Diagnostic Program (CommChk.exe)

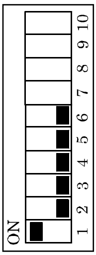
Performs actual communications and indicates whether the results are correct or not (error).

The following describes the procedure for testing using the serial communications diagnostic program (CommChk.exe).

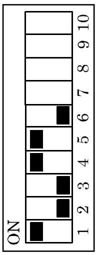
Check Method

Obtain an RS-422A/485 cross cable. If you do not have a cross cable, you can use a switch on the board to perform testing of a single COM port using loopback communications. See the figure below for the switch settings.

Switch setting for using a cross cable



Switch settings for using loopback communications (without a cross cable)



Using the Diagnosis Program

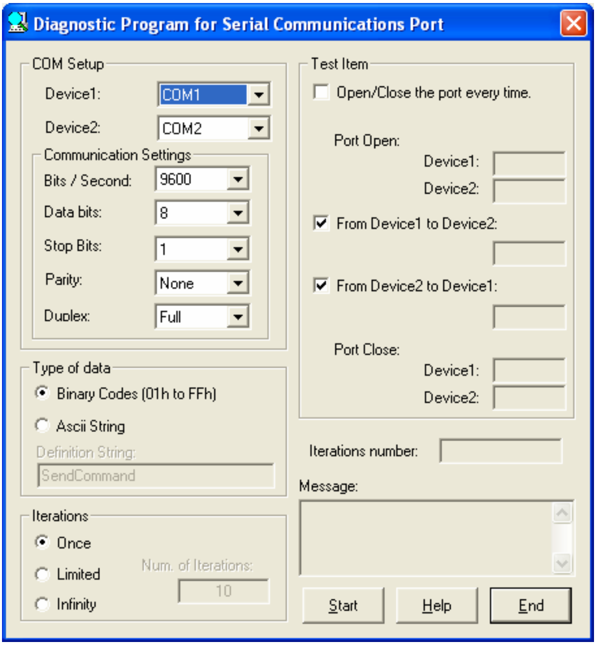
Starting the Diagnosis Program

Run\APISBP\Apisbp\ComDrv\Utility\CommChk\CommChk.exe from the supplied CD-ROM.

Communication Settings

COM Setup: Specify the number of the COM port you wish to test.
If connecting two COM ports via a cross cable, specify the respective COM ports in [Device1] and [Device2].
When performing loopback communications on a single COM port, set the same port number in both [Device 1] and [Device 2].

Communication Settings: Specify the [Bits / Second], [Data bits] and other settings you wish to use.

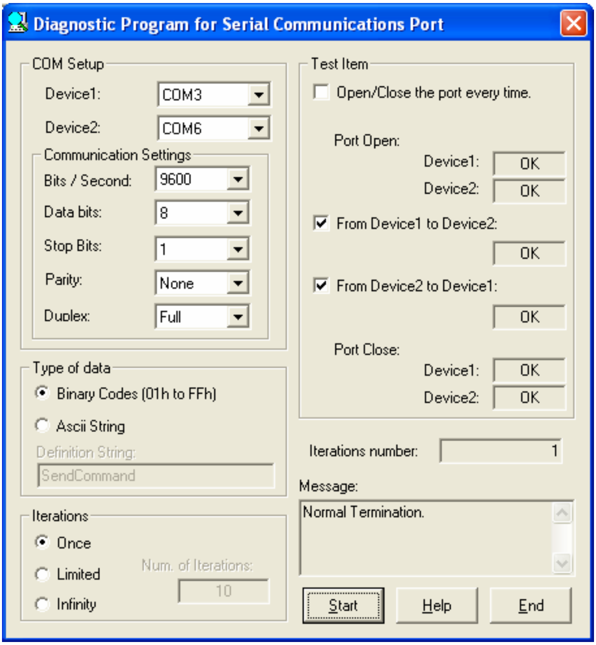


Start test

Click the [Start] button to start the test using the specified conditions.

View test result

The test result is displayed in the [Message] window.
A successful completion message appears if the test completed OK.



I/O Ports and Registers

The module uses the XR16C2850 (Exar Corporation) upward compatible with the 16550 UART.

For details on the internal registers of the XR16C2850 and its control, refer to the XR16C2850 data sheet. In enhanced mode, “starting I/O address + 1Fh” is used as an interrupt vector register.

I/O Ports

Starting I/O address	D7	D6	D5	D4	D3	D2	D1	D0
+0h – +7h	Channel 0 register							
+1Fh	Interrupt vector register							

Figure 5.1. I/O Ports

⚠ CAUTION

Each port must be accessed in bytes to conform to device specifications.
Non-byte access to any port is not allowed (such as word access or doubleword access).

Details on I/O Ports and Registers

The XR16C2850's internal registers are port-mapped differently depending on the value set in the Line Control Register (LCR).

The General Registers are enabled at startup or with LCR Bit 7 = 0 and LCR = other than 0xBFh.

The Baud Rate Registers are enabled with LCR Bit 7 = 1.

The Enhanced Registers are enabled with LCR = 0xBFh.

Input port 1 (General Registers)

Starting I/O address	D7	D6	D5	D4	D3	D2	D1	D0
+0h	Receive Holding Register (RHR)							
	Data Bit 7	Data Bit 6	Data Bit 5	Data Bit 4	Data Bit 3	Data Bit 2	Data Bit 1	Data Bit 0
+1h	Interrupt Enable Register (IER)							
	CTS Interrupt (*2)	RTS Interrupt (*2)	Xoff Interrupt (*2)	Sleep Mode (*2)	MODEM Status Interrupt	Receive Line Status Interrupt	Transmit Holding Register	Receive Holding Register
+2h	Interrupt Status Register (ISR)							
	FIFOs Enable	FIFOs Enable	-RTS, -CTS	Xoff	Int Priority Bit 2	Int Priority Bit 1	Int Priority Bit 0	Int Status
+3h	Line Control Register (LCR)							
	Divisor Latch Enable	Set Break	Set Parity	Even Parity	Parity Enable	Stop Bits	Word Length Bit 1	Word Length Bit 0
+4h	Modem Control Register (MCR)							
	Clock Prescaler Select (*2)	IR Mode Enable (*2)	Xon Any (*2)	Loop Back Enable	-OP2 and INT Enable	Out 1 0:INT Enable 1:INT Disable (*1)	-RTS	-DTR
+5h	Line Status Register (LSR)							
	FIFO Error	THR& TSR Empty	THR Empty	Break Interrupt	Framing Error	Parity Error	Overrun Error	Receive Data Ready
+6h	Modem Status Register (MSR)							
	-CD	-RI	-DSR	-CTS	Delta -CD	Delta -RI	Delta -DSR	Data -CTS
+7h	Scratch Pad Register (SPR)							
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
+7h	FIFO Level Counter (FLVL) (*3)							
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

*1 For details, see the "Using an Interrupt" section.

*2 These bits are enabled with EFR Bit 4 = 1.

*3 These registers are enabled with FCTR Bit 6 = 1.

Figure 5.2. Input port (General Registers)

Input port 2 (Baud Rate Registers)

Starting I/O address	D7	D6	D5	D4	D3	D2	D1	D0
+0h	LSB of Divisor Latch (DLL)							
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
+1h	MSB of Divisor Latch (DLM)							
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8

Figure 5.3. Input port (Baud Rate Registers)

Input port 3(Enhanced Registers)

Starting I/O address	D7	D6	D5	D4	D3	D2	D1	D0
+0h	FIFO Level Counter							
	Trig/FC	Trig/FC	Trig/FC	Trig/FC	Trig/FC	Trig/FC	Trig/FC	Trig/FC
+1h	Feature Control Register (FCTR)							
	Rx/Tx Mode	SCPAD Swap	Trig Bit 1	Trig Bit 0	RS485 Auto control	IrRx Inv.	-RTS Delay Bit 1	-RTS Delay Bit 0
+2h	Enhanced Feature Register (EFR)							
	Auto -CTS	Auto -RTS	Special Char, select	Enable IER Bits 4-7, ISR, FCR Bits 4-5, MCR Bits 5-7	Cont-3 Tx, Rx Control	Cont-2 Tx, Rx Control	Cont-1 Tx, Rx Control	Cont-0 Tx, Rx Control
+4h	Xon-1 Word							
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
+5h	Xon-2 Word							
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
+6h	Xoff-1 Word							
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
+7h	Xoff-2 Word							
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8

Figure 5.4. Input port (Enhanced Registers)

Input port 4 (Interrupt vector register)

Starting I/O address	D7	D6	D5	D4	D3	D2	D1	D0
+1Fh	Interrupt vector register							
	Fixed at "0"	(Reserved)						CH0 Interrupt

Figure 5.5. Input port (Interrupt vector register)

Output port 1(General Registers)

Starting I/O address	D7	D6	D5	D4	D3	D2	D1	D0
+0h	Transmit Holding Register (THR)							
	Data Bit 7	Data Bit 6	Data Bit 5	Data Bit 4	Data Bit 3	Data Bit 2	Data Bit 1	Data Bit 0
+1h	Interrupt Enable Register (IER)							
	CTS Interrupt (*2)	RTS Interrupt (*2)	Xoff Interrupt (*2)	Sleep Mode (*2)	MODEM Status Interrupt	Receive Line Status Interrupt	Transmit Holding Register	Receive Holding Register
+2h	FIFO Control Register (FCR)							
	RCVR	RCVR	TX Trigger (MSB)	TX Trigger (LSB)	DMA Mode Select	XMIT FIFO Reset	RCVR FIFO Reset	FIFO Enable
+3h	Line Control Register (LCR)							
	Divisor Latch Enable	Set Break	Set Parity	Even Parity	Parity Enable	Stop Bits	Word Length Bit 1	Word Length Bit 0
+4h	Modem Control Register (MCR)							
	Clock Prescaler Select (*2)	IR Mode Enable (*2)	Xon Any (*2)	Loop Back Enable	-OP2 and INT Enable	Out 1 0:INT Enable 1:INT Disable (*1)	-RTS	-DTR
+5h	Disable							
+6h	Disabled							
+7h	Scratch Pad Register (SPR)							
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
+7h	Enhanced Mode Select Registers (EMSR) (*3)							
	Reserved	Reserved	RTS Hyst Bit 3	RTS Hyst Bit 2	Reserved	Reserved	Alt.RX/TX FIFO Count	RX/TX FIFO Count

*1 For details, see the “Using an Interrupt” section.

*2 These bits are enabled with EFR Bit 4 = 1.

*3 These registers are enabled with FCTR Bit 6 = 1.

Figure 5.6. Output port (General Registers)

Output port 2(Baud Rate Registers)

Starting I/O address	D7	D6	D5	D4	D3	D2	D1	D0
+0h	LSB of Divisor Latch (DLL)							
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
+1h	MSB of Divisor Latch (DLM)							
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8

Figure 5.7. Output port (Baud Rate Registers)

Output port 3(Enhanced Registers)

Starting I/O address	D7	D6	D5	D4	D3	D2	D1	D0
+0h	FIFO Level Counter							
	Trig/FC	Trig/FC	Trig/FC	Trig/FC	Trig/FC	Trig/FC	Trig/FC	Trig/FC
+1h	Feature Control Register (FCTR)							
	Rx/Tx Mode	SCPAD Swap	Trig Bit 1	Trig Bit 0	RS485 Auto control	IrRx Inv.	-RTS Delay Bit 1	-RTS Delay Bit 0
+2h	Enhanced Feature Register (EFR)							
	Auto -CTS	Auto -RTS	Special Char, select	Enable IER Bits 4-7, ISR, FCR Bits 4-5, MCR Bits 5-7	Cont-3 Tx, Rx Control	Cont-2 Tx, Rx Control	Cont-1 Tx, Rx Control	Cont-0 Tx, Rx Control
+4h	Xon-1 Word							
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
+5h	Xon-2 Word							
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
+6h	Xoff-1 Word							
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
+7h	Xoff-2 Word							
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8

Figure 5.8. Output port (Enhanced Register)

Output port 4(Interrupt Vector Registers)

Starting I/O address	D7	D6	D5	D4	D3	D2	D1	D0
+1Fh	Interrupt Vector Registers							
	Fixed at "0"	(Reserved)					Clock selection Bit 1	Clock selection Bit 0

Figure 5.9. Output port (Interrupt Vector Registers)

Interrupt Vector Registers

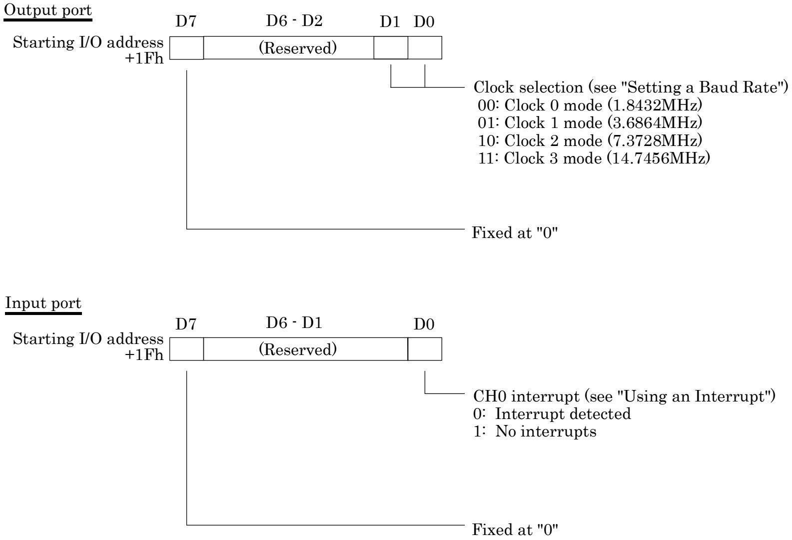


Figure 5.10. Interrupt Vector Registers

Using an Interrupt

Compatible Mode

When the Module is operated in the compatible mode, channel CH0 is assigned to standard I/O addresses COM3 and COM4, respectively. The interrupt levels will be IRQ5 for COM3 and IRQ7 for COM4.

Enhanced Mode

When the Module is being used in enhanced mode, interrupt generation can be checked with the interrupt vector register (IVR). The Device ID switch is used to specify the interrupt line to be used.

When accepting an interrupt, you can check, by reading the IVR, whether the interrupt has been generated within the interrupt service routine. Before exiting from the executed interrupt service, read the IVR again to check for any pending interrupt.

The XR16C2850 contains an internal register to enable itself for interrupts (make them available) or to check whether any interrupt has been generated. For using the register, refer to the data sheet for the XR16C2850 supplied by Exar Corporation.

For setting to use interrupts, use the D2 bit in the modem control register (MCR).

D2 bit of MCR	Write 0 : Enable(when the power is turned on)
	Write 1 : Disable

Examples

Following is a COM-1PD(FIT)GY sample program.

The sample program executes transmission and reception between two COM-1PD(FIT)GY units. To connect RTS and CTS to an external device in full-duplex mode, use a connection cable as illustrated below.

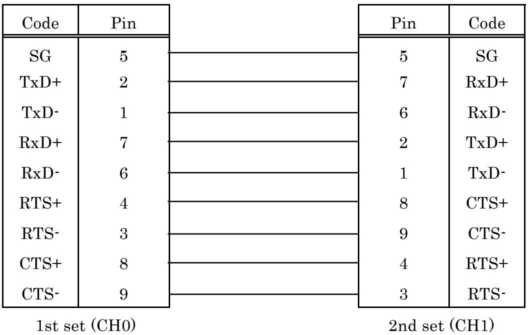
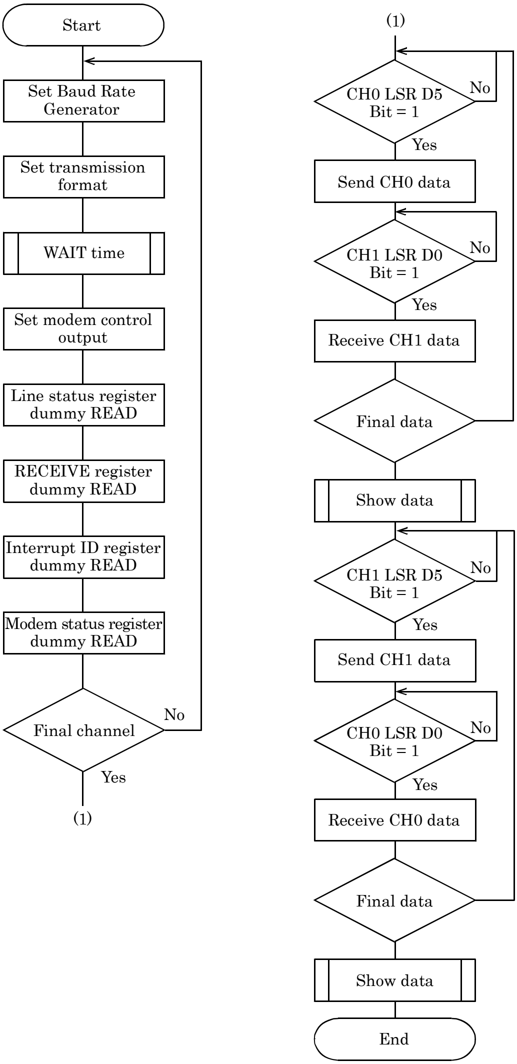


Figure 5.11.   Cable Connection Diagram

Compatible Mode

Flowchart



Sample Program

```
/*=========================================================================
   Sample program 1   -- UNIT0 <=> UNIT1 Loop Test --

      UNIT No.        0           1
      DEVICE ID:      0           2
      Mode:           Compatible Mode
      Channel 0:      03E8H (COM3) 02E8H (COM4)
      Baud Rate:      115,200 bps
      Interrupt:      N/A

========================================================================== */
#include <stdio.h>
#include <conio.h>

/* ----- Constant ------------------------------------------------------- */
#define   CH       2                            /* no. of channels */
#define   NUM      10                           /* data of total number */
#define   BPS      1                            /* baud rate */

         unsigned char   rbuf[CH][NUM];         /* receive buffer */
         unsigned int ADR[CH] = {0x3e8, 0x2e8}; /* I/O address */
         unsigned int rcnt[CH] = {0, 0};        /* receive count */


/* ----- Prototype ------------------------------------------------------ */
void   main( void );


/* ----- Main ----------------------------------------------------------- */
void   main( void )
{
   unsigned char      i, j;
   int             dummy;

   for(i = 0; i < CH; i ++) {            /* RS-232C initialize */
      outp( ADR[i]+0x3, 0x80 );        /* Line Control Register (DLAB SET) */
      outp( ADR[i]+0x0, BPS & 0xff ); /* Divisor Latch (Baud Rate SET) */
      outp( ADR[i]+0x1, (BPS >> 8) & 0xff );
      outp( ADR[i]+0x3, 0x0 );         /* Line Control Register (DLAB RESET) */
      outp( ADR[i]+0x3, 0x7 );         /* Line Control Register */
                                       /* ( Bit 0,1 ... Word Length = 8 bit ) */
                                       /* ( Bit 2 ..... 2 stop bit         ) */
      for(j = 0; j < 10; j++)          /* hardware wait */
         dummy = inp( 0x2ef );

      outp( ADR[i]+0x4, 0x3 );      /* MODEM Control Register (DTR . RTS) */
      dummy = inp( ADR[i]+0x5 );    /* Line Status Register(Dummy Read) */
      dummy = inp( ADR[i]+0x0 );    /* Receiver Buffer Register (Dummy Read) */
      dummy = inp( ADR[i]+0x2 );    /* Interrupt ID. Register (Dummy Read) */
      dummy = inp( ADR[i]+0x6 );    /* Modem Status Register (Dummy Read) */
      outp( ADR[i]+0x1, 0x1 );      /* Interrupt Enable Register */
   }
   for(j = 0; j < NUM; j++) {                   /* send CH0 to CH1 */
      while( !(inp( ADR[0]+0x5 ) & 0x20) );     /* transmitter empty */
      outp( ADR[0]+0x0, j );
      while( !(inp( ADR[1]+0x5 ) & 0x1) );      /* data ready */
      rbuf[1][rcnt[1]] = (unsigned char)inp( ADR[1]+0x0 );
      rcnt[1]++;
   }
```

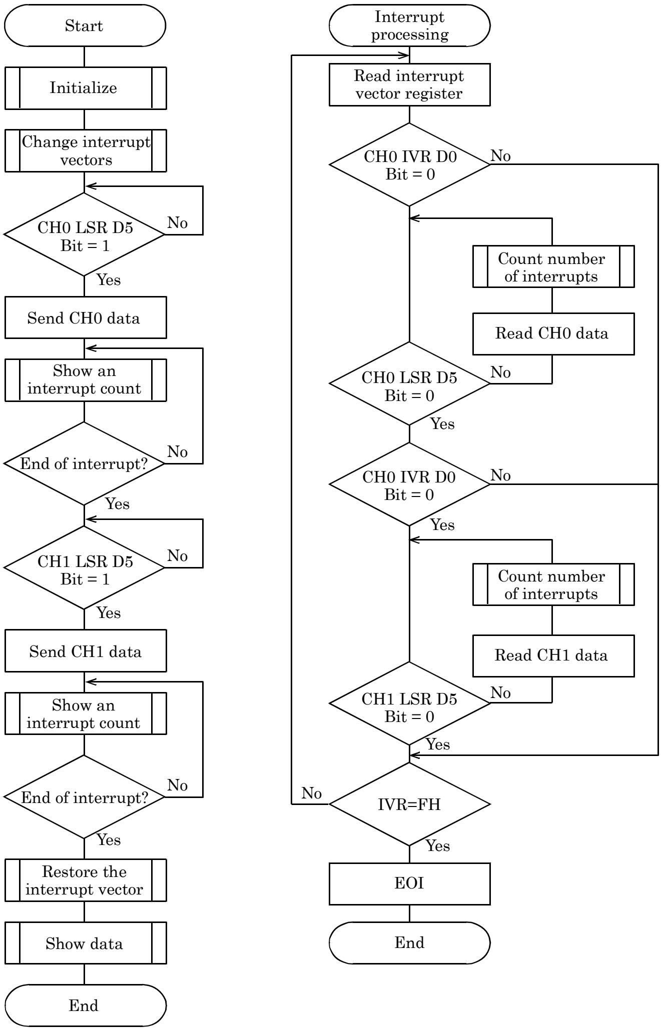
```
    for(j = 0; j < NUM; j++) {
        printf(" CH0:%02x  ->  CH1:%02x   ", j, rbuf[1][j]);
        if(rbuf[1][j] != j) printf("Verify Error \n");      /* verify data */
        else   printf("\n");
    }
    printf("\n");

    for(j = 0; j < NUM; j++) {                          /* send CH1 to CH0 */
        while( !(inp( ADR[1]+0x5 ) & 0x20) );           /* transmitter empty */
        outp( ADR[1]+0x0, j );
        while( !(inp( ADR[0]+0x5 ) & 0x1) );            /* data ready */
        rbuf[0][rcnt[0]] = (unsigned char)inp( ADR[0]+0x0 );
        rcnt[0]++;
    }
    for(j = 0; j < NUM; j++) {
        printf(" CH1:%02x  ->  CH0:%02x   ", j, rbuf[0][j]);
        if(rbuf[0][j] != j) printf("Verify Error \n");      /* verify data */
        else   printf("\n");
    }
}

/* ---------------------------------------------------------- End of file --- */
```

Enhanced Mode

Flowchart



Sample Program

```
/*
===========================================================================
   Sample program 2   -- UNIT0 <=> UNIT1 Loop Test --

      UNIT No.        0            1
      DEVICE ID:      4            7
      Mode:           Enhanced Mode
      Channel 0:      1A0H         2A0H
      Baud Rate:      115,200 bps
      Interrupt:      IRQ5         IRQ7

 ==========================================================================*/
#include <stdio.h>
#include <conio.h>
#include <dos.h>

/* ----- Constant ---------------------------------------------------------- */
#define  CH      2                            /* no. of channels */
#define  NUM     10                           /* data of total number */
#define  BPS     1                            /* baud rate */
#define  IRQ5    0                            /* IRQ5 */
#define  IRQ7    1                            /* IRQ7 */

volatile  unsigned char   rbuf[CH][NUM];        /* receive buffer */
volatile  unsigned int ADR[CH] = {0x1a0, 0x2a0}; /* I/O address */
volatile  unsigned int intcnt[CH] = {0, 0};     /* interrupt counter */
volatile  int          IrqLevel = IRQ5;         /* interrupt level */
          int          OrgMasterImr;            /* original IMR */
          unsigned char   IntVector[2] = { 0x0d, 0x0f };
                                                /* interruput vector */
          unsigned char   PicMask[2] = { 0xdf, 0x7f };
                                                /* mask bit */
          unsigned char   IsrClear[3] = { 0x65, 0x67, 0x61 };
                                                /* ISR clear */


/* ----- Prototype --------------------------------------------------------- */
void  main( void );
void  Initialize( void );                       /* initialize */
void  ChgVect( void );                          /* change vector */
void  ResVect( void );                          /* restore vector */
void  _interrupt _far inthandler( void );       /* interrupt handler */
void  ( _interrupt _far *OrgVect)();            /* original vector */

/* ----- Initialize -------------------------------------------------------- */
void  Initialize( void )
{
   unsigned int i, j;
   int             dummy;

   for(i = 0; i < CH; i++) {
      outp( ADR[i]+0x3, 0x80 );       /* Line Control Register (DLAB SET) */
      outp( ADR[i]+0x0, BPS & 0xff ); /* Divisor Latch (Baud Rate SET) */
      outp( ADR[i]+0x1, (BPS >> 8) & 0xff );
      outp( ADR[i]+0x3, 0x0 );        /* Line Control Register(DLAB RESET) */

      outp( ADR[i]+0x3, 0x7 );     /* Line Control Register */
                                   /*  ( Bit 0,1 ... Word Length = 8 bit ) */
                                   /*  ( Bit 2 ..... 2 stop bit          ) */
      for(j = 0; j < 10; j++)      /* hardware wait */
         dummy = inp( 0x2ef );
```

```
        outp( ADR[i]+0x4, 0x3 );   /* MODEM Control Register (DTR . RTS) */
        dummy = inp( ADR[i]+0x5 ); /* Line Status Register   (Dummy Read) */
        dummy = inp( ADR[i]+0x0 ); /* Receiver Buffer Register (Dummy Read) */
        dummy = inp( ADR[i]+0x2 ); /* Interrupt ID Register (Dummy Read) */
        dummy = inp( ADR[i]+0x6 ); /* Modem Status Register (Dummy Read) */
        outp( ADR[i]+0x1, 0x1 );   /* Interrupt Enable Register */
    }
    outp( ADR[0]+0x1f, 0x80 );    /* Select Clock Mode */
}


/* ----- change vector ----------------------------------------------------- */
void   ChgVect( void )
{
    OrgVect = _dos_getvect( IntVector[IRQ5] );
    OrgVect = _dos_getvect( IntVector[IRQ7] );
    _disable();
    _dos_setvect( IntVector[IRQ5], inthandler );
    _dos_setvect( IntVector[IRQ7], inthandler );
    outp( 0x21, ( OrgMasterImr = inp( 0x21 ) ) & PicMask[IRQ5] & PicMask[IRQ7] );
    outp( 0x20, IsrClear[IRQ5] );                         /* ISR clear */
    outp( 0x20, IsrClear[IRQ7] );                         /* ISR clear */
    _enable();                                            /* enable */
}


/* ----- restore vector ---------------------------------------------------- */
void   ResVect( void )
{
    _disable();                                    /* disable */
    outp( 0x21, OrgMasterImr );
    _dos_setvect( IntVector[IRQ5], OrgVect );      /* restore orgvect */
    _dos_setvect( IntVector[IRQ7], OrgVect );      /* restore orgvect */
    _enable();                                     /* enable */
}


/* ----- interrupt handler ------------------------------------------------- */
void   _interrupt _far inthandler( void )
{
    unsigned char      sts0, sts1;

    _enable();                                     /* enable */
    do {
        sts0 = (unsigned char)inp( ADR[0]+0x1f ); /* Int Vector Register */
        if( !(sts0 & 0x1) ) {                     /* CH0 status */
            while( inp( ADR[0]+0x05 ) & 0x1) {
                rbuf[0][intcnt[0]] = (unsigned char)inp( ADR[0]+0x0 );
                intcnt[0]++;                /* count times of an interrupt */
            }
        }
    } while( !(sts0 & 0xf) );

    do {
        sts1 = (unsigned char)inp( ADR[1]+0x1f ); /* Int Vector Register */
        if( !(sts1 & 0x1) ) {                     /* CH0 status */
            while( inp( ADR[1]+0x05 ) & 0x1) {
                rbuf[1][intcnt[1]] = (unsigned char)inp( ADR[1]+0x0 );
                intcnt[1]++;                /* count times of an interrupt */
            }
        }
    } while( !(sts1 & 0xf) );
```

```
    _disable();                                        /* disable */
    outp( 0x20, 0x20 );                                /* EOI */
}


/* ------ main ------------------------------------------------------------- */
void  main( void )
{
    unsigned int i, j;

    Initialize();                                       /* initialize */
    ChgVect();                                          /* change vector */

    for(j = 0; j < NUM; j++) {                         /* send CH0 to CH1 */
        while( !(inp( ADR[0]+0x5 ) & 0x20) );          /* transmitter empty */
        outp( ADR[0]+0x0, j );
    }
    while(intcnt[1] < 10)
        printf("Interrupt  CH0:%02d  CH1:%02d\n", intcnt[0], intcnt[1]);
    printf("\n");

    for(j = 0; j < NUM; j++) {                         /* send CH1 to CH0 */
        while( !(inp( ADR[1]+0x5 ) & 0x20) );          /* transmitter empty */
        outp( ADR[1]+0x0, j );
    }
    while(intcnt[0] < 10)
        printf("Interrupt  CH0:%02d  CH1:%02d\n", intcnt[0], intcnt[1]);
    printf("\n");

    ResVect();                                         /* restore vector */
    for(j = 0; j < NUM; j++) {
        printf(" CH0:%02x  ->  CH1:%02x   ", j, rbuf[1][j]);
        if(rbuf[1][j] != j) printf("Verify Error \n");      /* verify data */
        else  printf("\n");
    }
    printf("\n");
    for(j = 0; j < NUM; j++) {
        printf(" CH1:%02x  ->  CH0:%02x   ", j, rbuf[0][j]);
        if(rbuf[0][j] != j) printf("Verify Error \n");
        else  printf("\n");
    }
    printf("\n");
    printf("Interrupt  CH0:%02d  CH1:%02d\n", intcnt[0], intcnt[1]);
}

/* ------------------------------------------------------------ ---------------------------------End of file --- */
```

6. System Reference

Block Diagram

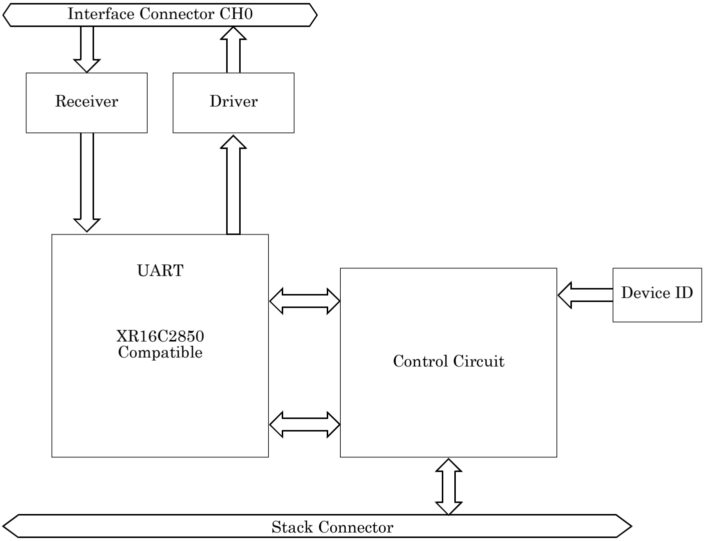


Figure 6.1. Block Diagram

Equivalence Circuit

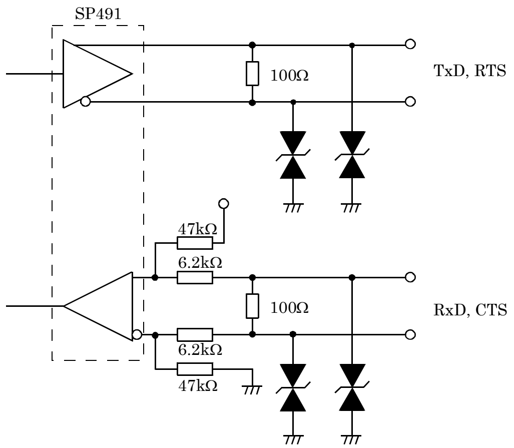


Figure 6.2. Circuitry Diagrams RS-422A/485 in Full Duplex

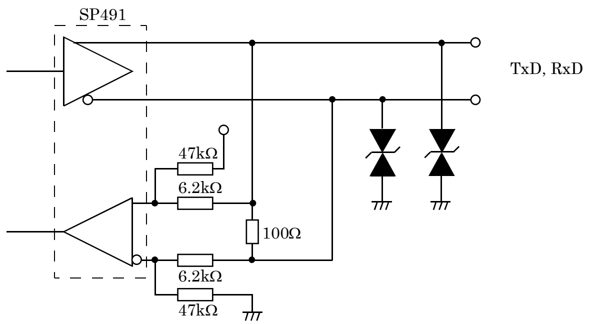


Figure 6.3. Circuitry Diagrams RS-422A/485 in Half Duplex

Specification

Table 6.1. Specification

Item	Specification
Number of channels	1ch
Interface type	RS-422A/RS-485
Isolation	Bus Isolation
Isolation voltage	1000VDC
Transfer method	Asynchronous serial transfer (Full/Half duplex)
Baud rate	2 - 921,600bps *1 *2
Data length	5, 6, 7, 8 bits 1, 1.5, 2 stop bits *1
Parity check	Even, Odd, Non-parity *1
Controller chip	162850 or equivalent (The module has 128-byte receive and 128-byte transmit FIFO buffers.)
Interrupt requests	1 level use
Power consumption	5VDC 300mA (Max.)
Connecting distance	Within 1200m *3
Dimension (mm)	25.2(W) x 64.7(D) x 94.0(H) (No protrusions)
Weight(module itself)	100g
Module connection method	Stack connection by the connector that is provided with the side of module
Module installation method	One-touch connection to 35mm DIN rails. (standard connection mechanism provided in the system)

*1 These items can be set by software.
*2 Data transmission at high speed may not be performed normally depending on the environment including the type of status of connected material of cable and environment.
*3 The table below lists an example of the relationship between baud rate and communication distance.

Communication distance	Baud rate
300m	115,200bps
600m	57,600bps
900m	19,200bps
1200m	9,600bps

Communication cable: 28AWG, double shielded cable, twisted pairs used for each +/- signal line.

⚠ CAUTION

When connecting the module to a controller module, the internal power consumption should be taken into account. If the total current exceeds the capacity of the power supply unit, the integrity of the operation cannot be guaranteed. For further details, please see the Controller Module manual.

Table 6.2. Installation Environment Requirements

Parameter	Requirement description
Operating temperature	0 - 50°C
Storage temperature	-10 - 60°C
Humidity	10 - 90% (No condensation)
Floating dust particles	Not to be excessive
Corrosive gases	None

Physical Dimensions

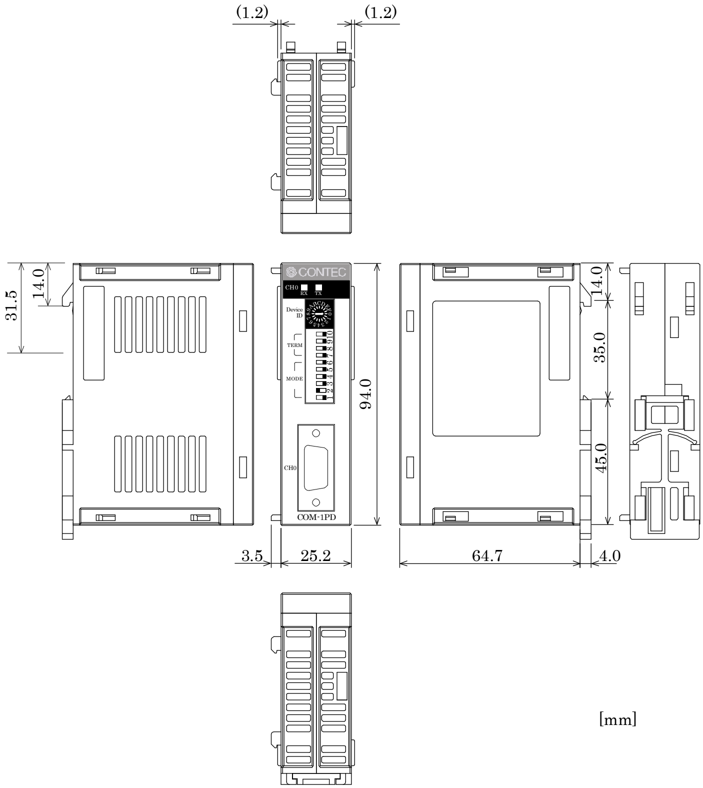


Figure 6.4. Physical Dimensions

COM-1PD(FIT)GY

User's Manual

CONTEC CO., LTD.

March 2009 Edition

3-9-31, Himesato, Nishiyodogawa-ku, Osaka 555-0025, Japan

Japanese http://www.contec.co.jp/
English http://www.contec.com/
Chinese http://www.contec.com.cn/

[12102003] Management No. A-46-812
[03192009_rev7] Parts No. LYDC896